Государственное ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ   
ВЫСШЕГО ПРОФЕССИОНАЛЬНОГО ОБРАЗОВАНИЯ

«Донецкий национальный технический университет»

Д09.03.04-ПОИС.20-20/5847.ЛР7

***Кафедра*** искусственного интеллекта  
 и системного анализа

Лабораторная работа №2

по дисциплине "Основы программирования"

на тему: "Работа с массивами"

Проверили:

\_\_\_\_\_\_\_\_\_\_\_\_ асс. А.П. Семёнова

(дата, подпись)

\_\_\_\_\_\_\_\_\_\_\_\_ асс. И.В. Савицкая

(дата, подпись)

Выполнил:

\_\_\_\_\_\_\_\_\_\_\_\_ ст.гр.ПИ-20г М. А. Евсеев

(дата, подпись)

Донецк – 2020

## Задание

1) Составить метод решения задачи и алгоритм обработки массива, в соответствии с заданием.

2) Написать программу на языке Си.

Вариант 9

Соседями элемента А[i,j] в матрице назовем элементы, расположенные рядом с данным элементом в строке и столбце. Провести операцию сглаживания, которая дает новую матрицу того же размера, каждый элемент которой - среднее арифметическое имеющихся соседей соответствующего элемента исходной матрицы.

# Ход работы

## Постановка Задачи

Исходя из матрицы необходимо вывести новый массив того же размера, элементы которого будут являться средним арифметическим имеющихся соседей соответствующего элемента исходной матрицы.

Входные данные:

arr [N][M] – целочисленный массив.

Промежуточные данные:

N, M – целочисленные константы, количество строк и столбцов;

i, j – целочисленное числа, счетчик цикла;

k – целочисленное число, счетчик соседей элемента.

Выходные данные:

new\_arr[N][M] – вещественный массив.

# Метод решения задачи

Введем элементы исходной матрицы. После чего, проверяя существование соседей элемента исходной матрицы, присвоим элементу новой матрицы их сумму. Зная количество соседней элемента и их сумму, найдем среднее арифметическое соседей элемента матрицы путем деления суммы на количество слагаемых. Выведем элемент новой матрицы ‑ результат.

# Алгоритм решения задачи

arr[i][j]

i=0;i<N;i++

j=0;j<M;j++

N-i==1

k=0

new\_arr[i][j]=0

i=0;i<N;i++

j=0;j<M;j++

M-j==1

i==0

j==0

new\_arr[i][j]+= arr[i-1][j]

k++

new\_arr[i][j]+= arr[i][j-1]

k++

new\_arr[i][j]+= arr[i+1][j]

k++

new\_arr[i][j]+= arr[i][j+1]

k++

new\_arr[i][j] = (float) new\_arr[i][j]/k;

new\_arr[i][j]

Да

Да

Да

Да

Нет

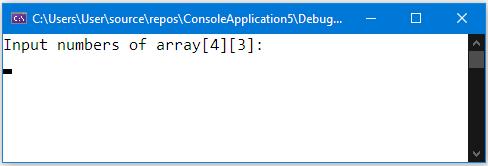
Нет

Нет

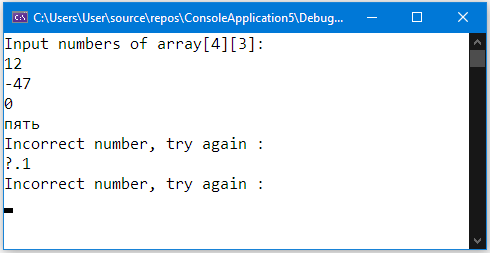
Нет

# Контрольные примеры

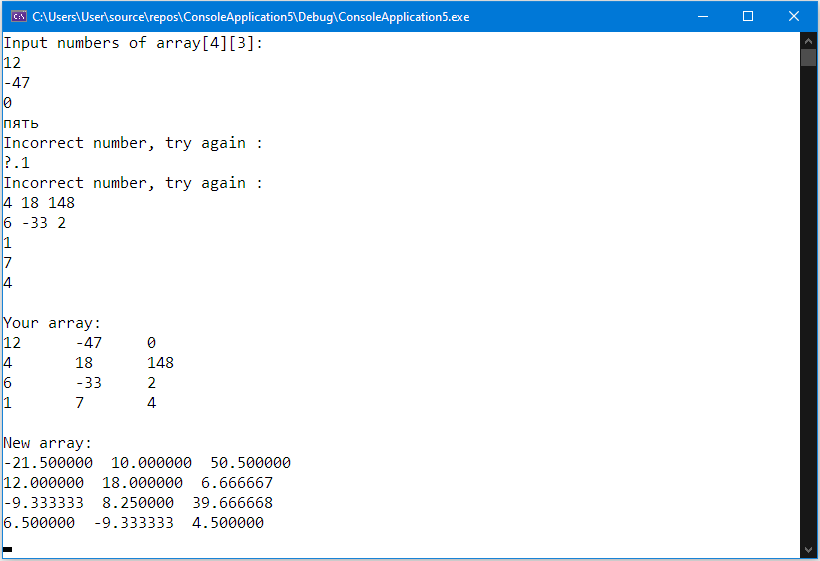
Запустив программу, мы должны ввести элементы матрицы:



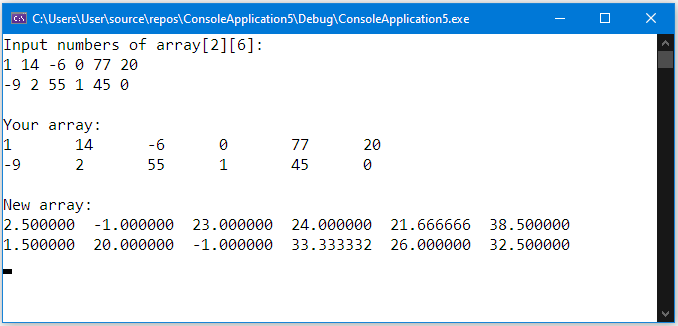
При некорректном вводе программа потребует повторный ввод элемента матрицы:



Введя элементы исходного массива, программа проведет операцию сглаживания и выдаст элементы нового массива, которые равны среднему арифметическому соседей элемента исходной матрицы :



Пример сглаживания матрицы размерностью 2х6:



Чтобы закрыть программу достаточно нажать любую клавишу.

# Листинг программы

Код программы, созданной в Visual Studio 2019:

#include <stdio.h>

#define N 2 //количество строк матриц

#define M 6 //количество столбцов матриц

//Очищаем поток входных данных

**void** flush\_input(**void**)

{

**char** c;

**while** (scanf("%c", &c) == 1 && c != '\n');

}

**int** main()

{

**int** arr[N][M], i, j, k;

//вводим элементы исходного массива

printf("Input numbers of array[%d][%d]:\n", N, M);

**for** (i = 0; i < N; i++)

{

**for** (j = 0; j < M; j++)

{

**while** (scanf("%d", &arr[i][j]) == 0)//обработка в случае неверного ввода числа

{

printf("Incorrect number, try again : \n");

flush\_input();

}

}

}

printf("\nYour array:\n");

**for** (i = 0; i < N; i++)

{

**for** (j = 0; j < M; j++)

{

printf("%d\t", arr[i][j]);

}

printf("\n");

}

//Операция сглаживания матрицы:

**float** new\_arr[N][M]; //инициализация нового массива

printf("\nNew array:\n");

**for** (i = 0; i < N; i++)

{

**for** (j = 0; j < M; j++)

{

k = 0;

new\_arr[i][j] = 0;

//проверки на существование соседей элемента матрицы

**if** (i != 0)

{

k += 1;

new\_arr[i][j] += arr[i - 1][j];

}

**if** (N - i != 1)

{

k += 1;

new\_arr[i][j] += arr[i + 1][j];

}

**if** (j != 0)

{

k += 1;

new\_arr[i][j] += arr[i][j - 1];

}

**if** (M - j != 1)

{

k += 1;

new\_arr[i][j] += arr[i][j + 1];

}

new\_arr[i][j] = (**float**)new\_arr[i][j] / k;

printf("%f ", new\_arr[i][j]);

}

printf("\n");

}

getch();

}

Вывод: Я научился cоставлять методы решения задачи и алгоритмы обработки массива, писать программы на языке Си для решения этих задач, тестировать их и обрабатывать исключения.